Primeiramente vamos utilizar o trello por se tratar de uma ferramenta gratuita e amplamente utilizada pela comunidade, ela nos permite gerenciar o nosso projeto com a nossa equipe, separando o nosso fluxo de tarefas ou o monitoramento das mesmas. É uma ferramenta muito completa, gratuita e também muito visual se tornando a escolha perfeita para a nossa equipe.

Para a nossa comunicação também vamos partir para uma ferramenta gratuita, porém muito completa. Vamos utilizar o discord, uma ferramenta de comunicação muito utilizada por vários setores, e também pode ser utilizada para uma equipe de desenvolvimento, pois conseguimos separar canais de voz para comunicação, assim podendo deixar os times separados. Além de conter canais de textos, alertar, e possuir Bots para automatizar mensagens entre outras coisas.

Para gestão de conhecimento resolvi adotar um Flarum por ser um software de código aberto e é um software de fórum online para comunicação. É uma importante ferramenta para o nosso Stack devido a comunidade ativa de devs que o utilizam, e pela facilidade de uso.

Para nosso sistema de controle de versão resolvi utilizar o git, pois é o sistema mais utilizado pelos desenvolvedores atualmente, é código aberto, e tem manutenção ativa, além de possuir um ótimo controle de fluxo de trabalho com as branchs.

O software de integração continua utilizado será o Jenkins, já que estamos focando em criar uma Stack alocando pouco recurso monetário quando possível é uma ótima escolha, já que é um servidor de código aberto. A principal função será gerenciar toda a nossa pipeline de entrega de software, controlar os processos de entrega em todos os ciclos do nosso software, desde a documentação até os testes e implantação. Além disso ele funciona muito bem com outros softwares da nossa Stack.

Para fazer o controle de nossas builds e automatiza-las eu escolhi utilizar mais um software de código aberto, o Maven. Com o Maven vamos conseguir gerenciar as nossas dependências do projeto java facilitando assim as nossas configurações de dependências do projeto.

Para o controle de database, utilizaremos o Flyway, ferramenta que permite controlar e gerenciar as mudanças do nosso banco de dados, com ele todas as nossas mudanças podem ser controladas e versionadas, além de automatizar execuções de scripts. O Flyway além de tudo também permite integrações com outras ferramentas do nosso Stack como o Maven, sendo muito voltada para ambientes em java.

A nossa ferramenta de testes em nosso Stack, foi escolhido o jUnit, uma ferramenta de código aberto. O jUnit nos permite construir testes automatizados em java e assim nos apresentando os resultados assim facilitando a criação e manutenção do nosso codigo.

Nossa ferramenta de entrega continua utilizada será o Spinnaker, outra ferramenta open-source para que possamos minimizar nossos gastos, porém mantar qualidade na nossa Stack. Com o Spinnaker conseguimos criar uma pipeline de entrega, automatizando a execução de diversas tarefas, com isso temos velocidade e segurança. Além disso funciona muito bem utilizando outras ferramentas da nossa Stack como Jenkins.

Para nossas configurações de infraestrutura resolvi adotar o terraform. Terraform hoje em dia é uma das ferramentas mais utilizadas para infraestrutura, ela vai servir no nosso Stack para gerenciar a infraestrutura de nuvem, além de ela permitir o gerenciamento de qualquer tipo de infraestrutura ela também é open source.

Para gerenciar os nossos artefatos, vamos utilizar o Jfrog no nosso Stack, com ele vamos acelerar ainda mais as nossas implantações de software. Além de tudo ele implementa muito bem com AWS, outra ferramenta que será utilizável na nossa Stack. Ressaltando que este também é uma ferramenta open-source.

Iremos utilizar AWS como nossa plataforma Cloud. A plataforma AWS é uma das mais adotadas no mundo atualmente, e nos entrega uma grande quantidade de serviços, que só serão pagos se utilizados. A escolha da AWS é pela quantidade de recursos que nos oferece, pela promessa de reduzir custos para as empresas, e por ser uma ferramenta mais utilizada no mundo que cresce constantemente.

A ferramenta de orquestração de container utilizada será o Kubernetes, uma ferramenta open-source muito utilizada globalmente, com esta ferramenta vamos conseguir gerenciar e implantar os nossos aplicativos, fazer implantações automatizadas de nossas aplicações entre outras coisas, além de utilizar com AWS.

Para fazer nossos BIs, monitorar e nossos logs utilizaremos o Prometheus, que é uma ferramenta com código aberto, leve e eficiente, que contém diversos recursos de monitoramento para serem utilizados, além de emitir alertas quando ocorrem eventos indesejados. Prometheus além de nos proporcionar diversos recursos ele também é muito utilizado junto com ferramentas de orquestração de containers, como por exemplo o kubernetes, que também estamos usando na nossa Stack, estão pelo motivo de ser gratuito, conter diversos recursos e ainda se integrar a sistemas modernos, resolvi utilizar o Prometheus para finalizar a montagem da nossa Stack.